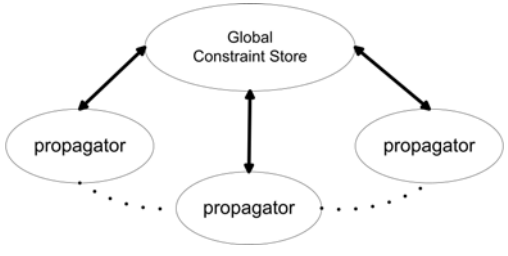
Embedding Constraint Satisfaction using Parallel Soft-Core Processors on FPGAs

Prasad Subramanian, Brandon Eames, Department of Electrical Engineering, Utah State University

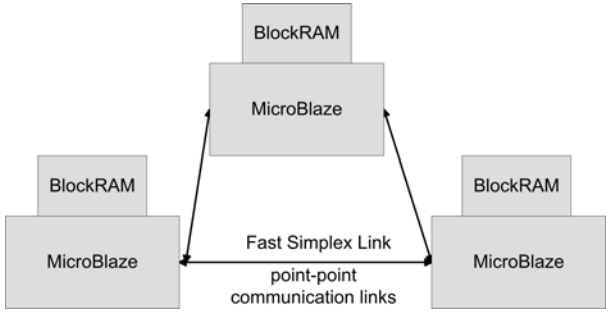
Abstract The architecture of a generic, flexible and scalable embedded finite domain constraint solver targeting an FPGA is presented in this work. We exploit the spatial parallelism offered by COTS FPGA architectures via the instantiation of multiple soft-core processors, which collectively implement the constraint solver. Soft-core processors facilitate the development of flexible software based algorithms for implementing individual constraints. The multi-core architecture realized on the FPGA facilitates tight inter-core synchronization required when solving constraints in parallel.

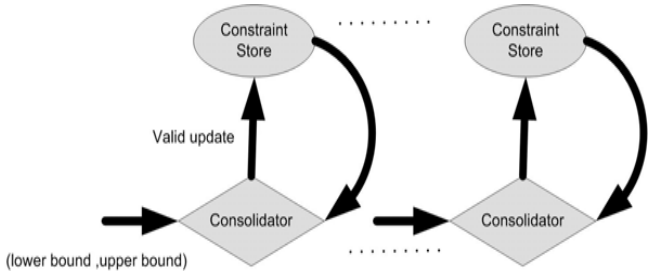
Introduction Constraint satisfaction and optimization techniques are commonly employed in scheduling problems, industrial manufacturing and automation processes, borrowing concepts from Operations Research (OR) and Artificial Intelligence (AI). Constraint satisfaction has also been applied in the design, synthesis and optimization of embedded systems. By modeling the scheduling problem as a constraint satisfaction problem (CSP), the embedded system becomes adaptable to dynamic changes in the environment. In this work, we model the task graph of dynamic occurrences for a space application in terms of precedence constraints and use the multi-processor embedded CSP solver.



Solver Architecture Propagation in a CSP with variables belonging to domain of finite cardinality is amenable to parallel processing. Instantiation of multiple soft-core MicroBlaze processors from Xilinx with distributed memories exploited the spatial parallelism provided by FPGA. The model in the above figure implies the need for a globally shared memory, simultaneously accessible by all propagation elements for sharing variable information. Without such sharing, propagators cannot cooperate to jointly make progress towards a solution.

Architecture The implementation of a low-latency, globally shared memory accessible by several computational devices is not practical on an FPGA fabric, due to limitations on the number of read-write ports of internal memories. Our implementation emulates shared memory by making use of on-chip BlockRAM. The constraint store is partitioned and distributed among the local memories associated with each soft core processor.





The Consolidator Emulation of shared memory via distributed memories and communication links imposes issues of coherency. Local copies of shared data are cached on each processor requiring. Data coherency is maintained through two steps. First, when a finite domain variable is updated by a remote processor, updates are sent to the owning processor. Second, all updates are routed through a hardware unit called a consolidator. The consolidator is a comparator that acts as a check point for all updates to the constraint store, and only accepts updates which improve or tighten the currently stored bounds for the variable. Since propagation approaches a solution through bounds analysis of finite domain variables, ordering between updates need not be maintained.

Results

To evaluate our embedded FD constraint solver, we used a task graph from a previously published hypothetical graph representing the events in an autonomous space mission planning algorithm. Our constraint model consists of enforcing temporal precedence constraints in order to derive a schedule of events in the graph. Measurements from the solver implementation executing on a VirtexII Pro Xilinx FPGA are provided in the table. Results indicate that a performance speed-up in propagation increases as the number of processors is increased. The solver failed to converge in the single-processor case, due to lack of sufficient space in the local configuration stack.

Clusters	Time to propagate(ticks)	# of distb. steps	First solution(ticks)	Cluster size	Propagation speed-up	Distb speed-up	% of stack usage
1	310209	60	FAILS	101	**1**	NA	167.98
2	159608	47	2632804	50/51	**1.94**	1.00	68.99
3	109971	43	1668505	34/34/33	**2.82**	1.58	44.41
4	85914	65	2360598	26/26/26/23	**3.61**	1.12	53.00

Conclusion

Online constraint satisfaction potentially opens the door to a variety of introspective dynamic optimizations to embedded systems. We have developed an approach for embedding a finite domain constraint solver on an FPGA, using a network of soft-core processors, distributed memories and point-to-point communication. The implementation of the solver framework is generic enough to allow different propagators embodying various other constraints to be added in the system.